Bootstrap and Jackknife Variance Estimators for the Pearson Correlation Coefficient

Stephen Agyeah

Introduction

This project investigates two resampling techniques *bootstrap* and *jackknife* for estimating the variance of the sample Pearson correlation coefficient. The study was motivated by the need to understand how these methods perform under different data distributions and sample sizes. Monte Carlo simulation was employed to provide empirical evidence and compare estimator performance.

Objectives

To implement *bootstrap* and *jackknife* methods for variance estimation.
To study the performance of both methods under: *Bivariate Normal Distribution Bivariate Lognormal Distribution* To compare variance estimates across different correlation values (ρ = 0, 0.2, 0.5, 0.8) and sample sizes (n = 15, 30).
To assess bias, consistency, and computational cost of the two methods.

Methodology

Simulation Design

Generated 10,000 Monte Carlo samples from the target distributions.
Considered sample sizes: **n = 15** and **n = 30**.
True correlation parameter ρ was varied (0, 0.2, 0.5, 0.8).

Resampling Approaches

Bootstrap: 1,000 bootstrap resamples drawn for each Monte Carlo sample to estimate variance.
Jackknife: Leave-one-out resampling applied to compute jackknife variance estimates.

Evaluation Metrics

Mean estimated variance
Standard error
Bias relative to the true Monte Carlo variance
Computational efficiency

BOOTSTRAP and Jackknife

This chunk defines the function simulate_bias_mse_all(), which runs Monte Carlo experiments to evaluate bias and mean squared error (MSE) of bootstrap variance estimates for the sample Pearson correlation.

Specifically, it:

Loops over two sample sizes (n = 15, 30) and four correlation values (ρ = 0, 0.2, 0.5,0.8) Simulates data from either a bivariate normal or bivariate lognormal distribution. *Computes the sample correlation for each simulated dataset.* Applies bootstrap resampling (B replicates) to estimate variance of the correlation. *Records Monte Carlo variance, bootstrap variance, bias, and MSE into a results data frame.

```r
library(MASS)    # load MASS package to access mvrnorm() for multivariate normal sampling


# function to compute bias and MSE for all combinations of n and rho
simulate_bias_mse_all <- function(m = 100, B = 10, dist = "normal") { # with dist normal

n_values <- c(15, 30)                # sample sizes to test
rho_values <- c(0, 0.2, 0.5, 0.8)    # correlation values to test
mu <- c(0, 0)                        # mean vector for MVN or lognormal

results <- data.frame()              # empty data frame to store results


# for loop over each sample size
for (n in n_values) {
for (rho in rho_values) {                        # for loop over each correlation

Sigma <- matrix(c(1, rho, rho, 1), ncol = 2)    # covariance matrix with given rho



r_vals <- numeric(m)            # store r values for Monte Carlo sample correlations
var_boot_vals <- numeric(m)     # store bootstrap variance estimates



# for Monte Carlo simulation
for (i in 1:m) {
```

```r
if (dist == "normal") {   # Check if the distribution type is normal
  Sigma <- matrix(c(1, rho, rho, 1), ncol = 2)# matrix with variances and
specific (rho)
  sample_data <- mvrnorm(n, mu, Sigma)    # bivariate normal sample of size n

} else {  # If the distribution is not 'normal', assume 'lognormal'
  sigma2 <- 1
  rho_log <- log(1 + rho * (exp(sigma2) - 1)) # Convert the desired
correlation in lognormal space

  Sigma <- matrix(c(1, rho_log, rho_log, 1), ncol = 2) # Create the
covariance matrix using the adjusted correlation
  sample_data <- exp(mvrnorm(n, mu, Sigma))  # Generate bivariate normal
samples, then exponentiate to obtain lognormal samples
}



x <- sample_data[, 1]                    # first variable x
y <- sample_data[, 2]                    # second variable y


r_vals[i] <- cor(x, y)                   # computing correlation from sample

# Bootstrap to estimate variance of r
r_boot <- numeric(B)                     # to store bootstrap r

for (b in 1:B) {
idx <- sample(1:n, replace = TRUE)       # resampling from the sample data
r_boot[b] <- cor(x[idx], y[idx])         # compute r on resampled data
        }
var_boot_vals[i] <- var(r_boot)          # computing variance of bootstrap
correlations
      }

# final calculations
var_mc <- var(r_vals)                     # estimated variance from MC
var_boot_mean <- mean(var_boot_vals)      # mean of bootstrap variance
estimates
bias <- var_boot_mean - var_mc            # bias mean(var_boot_vals) - var_mc
mse <- mean((var_boot_vals - var_mc)^2)  # MSE

# adding result to the output data frame
results <- rbind(results, data.frame(
distribution = dist,
n = n,
rho = rho,
var_mc = var_mc,
var_boot = var_boot_mean,
```

```
bias = bias,
mse = mse
      ))

cat( dist, "| n =", n, "| rho =", rho, "\n")
    }
  }

  return(results)
}
```

Bootstrap (Normal Distribution) Results

```
result_boot_normal <- simulate_bias_mse_all(m = 10000, B = 1000, dist =
"normal")

## normal | n = 15 | rho = 0
## normal | n = 15 | rho = 0.2
## normal | n = 15 | rho = 0.5
## normal | n = 15 | rho = 0.8
## normal | n = 30 | rho = 0
## normal | n = 30 | rho = 0.2
## normal | n = 30 | rho = 0.5
## normal | n = 30 | rho = 0.8

result_boot_normal

##   distribution  n rho      var_mc    var_boot          bias          mse
## 1       normal 15 0.0 0.071025725 0.063225331 -0.0078003936 6.890329e-04
## 2       normal 15 0.2 0.066865806 0.060062859 -0.0068029469 7.446263e-04
## 3       normal 15 0.5 0.044115753 0.043140679 -0.0009750732 6.428680e-04
## 4       normal 15 0.8 0.011548067 0.014256167  0.0027081002 2.069634e-04
## 5       normal 30 0.0 0.034532637 0.031442604 -0.0030900326 1.077268e-04
## 6       normal 30 0.2 0.031854678 0.029366780 -0.0024878981 1.056324e-04
## 7       normal 30 0.5 0.020355685 0.019664727 -0.0006909578 8.317088e-05
## 8       normal 30 0.8 0.005025486 0.005420392  0.0003949060 1.533027e-05
```

This table above shows variance estimation results for the Pearson correlation coefficient under a bivariate normal distribution using the bootstrap method with m = 10,000 simulations and B = 1000 bootstrap replications

As sample size increases from 15 to 30, the Monte Carlo variance (var_mc) consistently decreases confirming that larger samples lead to more stable estimates

Additionally,(the true correlation) also influences variance: When rho is close to 0, the correlation estimator tends to have higher variance. but as rho increases toward 1, the estimator becomes more stable, and its variance decreases

The bootstrap variance estimates (var_boot) are generally close to the true Monte Carlo variances, with very small bias in most cases

The bias values are typically small and negative, meaning the bootstrap slightly underestimates the variance in some cases

MSE values are all very low, indicating that the bootstrap is performing very accurately under the bivariate normal distribution

Bootstrap (Lognormal Distribution) Results

```
result_boot_lognormal <- simulate_bias_mse_all(m = 10000, B = 1000, dist =
"lognormal")

## lognormal | n = 15 | rho = 0
## lognormal | n = 15 | rho = 0.2
## lognormal | n = 15 | rho = 0.5
## lognormal | n = 15 | rho = 0.8
## lognormal | n = 30 | rho = 0
## lognormal | n = 30 | rho = 0.2
## lognormal | n = 30 | rho = 0.5
## lognormal | n = 30 | rho = 0.8

result_boot_lognormal

##   distribution  n rho     var_mc    var_boot         bias          mse
## 1    lognormal 15 0.0 0.07341380 0.063774965 -0.009638832 1.857742e-03
## 2    lognormal 15 0.2 0.08376657 0.066910625 -0.016855943 1.826200e-03
## 3    lognormal 15 0.5 0.06189851 0.048235399 -0.013663106 1.162771e-03
## 4    lognormal 15 0.8 0.01777043 0.015798458 -0.001971968 2.491266e-04
## 5    lognormal 30 0.0 0.03503212 0.029910619 -0.005121501 5.480922e-04
## 6    lognormal 30 0.2 0.04745466 0.035754264 -0.011700395 6.838333e-04
## 7    lognormal 30 0.5 0.03861938 0.027409799 -0.011209583 4.071808e-04
## 8    lognormal 30 0.8 0.01088195 0.008352703 -0.002529243 5.325661e-05
```

This table summarizes the variance estimation of the sample correlation coefficient using the bootstrap method under a bivariate lognormal distribution, with m = 10,000 simulations and B = 1000 bootstrap

Just like with the normal distribution, as sample size increases from 15 to 30, the Monte Carlo variance (var_mc) decreases, confirming that larger samples lead to more stable estimates

However, the bootstrap variance estimates (var_boot) are consistently lower than the true variance in all cases — leading to a negative bias across the board

The bias is more pronounced under the lognormal distribution than in the normal case, especially when the correlation rho is moderate (thus 0.2 or 0.5)

Combined Boot Results

```
combined_boot_results <- rbind(result_boot_normal, result_boot_lognormal)
combined_boot_results
```

```
##    distribution  n rho      var_mc    var_boot          bias          mse
## 1        normal 15 0.0 0.071025725 0.063225331 -0.0078003936 6.890329e-04
## 2        normal 15 0.2 0.066865806 0.060062859 -0.0068029469 7.446263e-04
## 3        normal 15 0.5 0.044115753 0.043140679 -0.0009750732 6.428680e-04
## 4        normal 15 0.8 0.011548067 0.014256167  0.0027081002 2.069634e-04
## 5        normal 30 0.0 0.034532637 0.031442604 -0.0030900326 1.077268e-04
## 6        normal 30 0.2 0.031854678 0.029366780 -0.0024878981 1.056324e-04
## 7        normal 30 0.5 0.020355685 0.019664727 -0.0006909578 8.317088e-05
## 8        normal 30 0.8 0.005025486 0.005420392  0.0003949060 1.533027e-05
## 9     lognormal 15 0.0 0.073413797 0.063774965 -0.0096388322 1.857742e-03
## 10    lognormal 15 0.2 0.083766568 0.066910625 -0.0168559426 1.826200e-03
## 11    lognormal 15 0.5 0.061898505 0.048235399 -0.0136631059 1.162771e-03
## 12    lognormal 15 0.8 0.017770427 0.015798458 -0.0019719685 2.491266e-04
## 13    lognormal 30 0.0 0.035032121 0.029910619 -0.0051215013 5.480922e-04
## 14    lognormal 30 0.2 0.047454658 0.035754264 -0.0117003948 6.838333e-04
## 15    lognormal 30 0.5 0.038619382 0.027409799 -0.0112095830 4.071808e-04
## 16    lognormal 30 0.8 0.010881946 0.008352703 -0.0025292427 5.325661e-05
```

JACKKNIFE

This chunk defines simulate_jackknife(), which runs a Monte Carlo study to evaluate the jackknife variance of the sample Pearson correlation and compare it to the Monte Carlo "true" variance.

It:

Loops over n={15, 30} and ρ={0, 0.2, 0.5, 0.8} for either bivariate normal or bivariate lognormal data. For each replicate, computes the sample correlation r and then performs leave-one-out jackknife to get n correlations. *Uses the classical jackknife variance formula* Aggregates across m simulations to report: Monte Carlo variance of r, mean jackknife variance, bias (jackknife − MC), and MSE. *Prints progress for each (dist, n, ρ) setting and returns a tidy results table

```r
library(MASS)


# function to simulate jackknife variance and compare to Monte Carlo variance
simulate_jackknife <- function(m = 10000, dist = "normal") {

n_values <- c(15, 30)                       # sample sizes to test
rho_values <- c(0, 0.2, 0.5, 0.8)           # correlation values to test
mu <- c(0, 0)                               # mean vector for MVN or lognormal
results <- data.frame()                     # empty data frame to store
results



# for loop over each sample size
for (n in n_values) {
```

```r
for (rho in rho_values) {                          # for loop over each
correlation
Sigma <- matrix(c(1, rho, rho, 1), ncol = 2)       # covariance matrix with
given rho

r_vals <- numeric(m)                    # store Monte Carlo r values
var_jack_vals <- numeric(m)             # store jackknife variance estimates


# for Monte Carlo simulation
for (i in 1:m) {

# sample from bivariate normal or lognormal
if (dist == "normal") {
  Sigma <- matrix(c(1, rho, rho, 1), ncol = 2)
  sample_data <- mvrnorm(n, mu, Sigma)

} else {
  sigma2 <- 1
  rho_log <- log(1 + rho * (exp(sigma2) - 1))
  Sigma <- matrix(c(1, rho_log, rho_log, 1), ncol = 2)
  sample_data <- exp(mvrnorm(n, mu, Sigma))
}

x <- sample_data[, 1]                # first variable x
y <- sample_data[, 2]                # second variable y

r_vals[i] <- cor(x, y)               # computing correlation from sample

# Jackknife variance estimation
r_jack <- numeric(n)
for (j in 1:n) {
r_jack[j] <- cor(x[-j], y[-j])  # Correlation with jth observation removed
}

# jackknife variance using classical formula
 r_bar <- mean(r_jack)
var_jack_vals[i] <- (n - 1) / n * sum((r_jack - r_bar)^2)
      }


var_mc <- var(r_vals)                   # compute Monte Carlo variance of r
var_jack_mean <- mean(var_jack_vals)    # mean of jackknife variance estimates

# adding result to the output data frame
results <- rbind(results, data.frame(
distribution = dist,
n = n,
```

```
rho = rho,
var_mc = var_mc,
var_jack = var_jack_mean,
bias = var_jack_mean - var_mc,
mse = mean((var_jack_vals - var_mc)^2)
      ))

cat(dist, "| n =", n, "| rho =", rho, "\n")
    }
  }
  return(results)
}
```

Jackknife (Normal Distribution) Results

```
result_jack_normal <- simulate_jackknife(m = 10000, dist = "normal")

## normal | n = 15 | rho = 0
## normal | n = 15 | rho = 0.2
## normal | n = 15 | rho = 0.5
## normal | n = 15 | rho = 0.8
## normal | n = 30 | rho = 0
## normal | n = 30 | rho = 0.2
## normal | n = 30 | rho = 0.5
## normal | n = 30 | rho = 0.8

result_jack_normal

##   distribution  n rho      var_mc    var_jack         bias          mse
## 1       normal 15 0.0 0.071239191 0.082910389 0.0116711984 2.706905e-03
## 2       normal 15 0.2 0.067606369 0.076919993 0.0093136240 2.358106e-03
## 3       normal 15 0.5 0.044455711 0.052844400 0.0083886894 1.819451e-03
## 4       normal 15 0.8 0.011655757 0.015601314 0.0039455568 3.954857e-04
## 5       normal 30 0.0 0.035433400 0.036710963 0.0012775626 2.047055e-04
## 6       normal 30 0.2 0.032376530 0.034184581 0.0018080511 2.125851e-04
## 7       normal 30 0.5 0.020356800 0.022288020 0.0019312202 1.494116e-04
## 8       normal 30 0.8 0.005071503 0.005747295 0.0006757922 2.116922e-05
```

Jackknife (Normal Distribution) Results

This table shows the jackknife-based variance estimation for the Pearson correlation coefficient under a bivariate normal distribution, using m = 10,000 Monte Carlo simulations

The jackknife estimates (var_jack) are slightly higher than var_mc, especially for smaller sample sizes resulting in positive bias in all scenarios

The bias is most noticeable for n = 15, particularly when rho is moderate (aorund 0.2 or 0.5), but becomes very small when n = 30

The MSE values are low across all settings, indicating that jackknife still performs reasonably well under normality

Under a bivariate normal distribution, the jackknife method provides reasonably accurate variance estimates of the sample correlation. It tends to slightly overestimate the variance in small samples, but the bias decreases substantially as the sample size increases

Jackknife (Lognormal Distribution) Results

```
result_jack_lognormal <- simulate_jackknife(m = 10000, dist = "lognormal")

## lognormal | n = 15 | rho = 0
## lognormal | n = 15 | rho = 0.2
## lognormal | n = 15 | rho = 0.5
## lognormal | n = 15 | rho = 0.8
## lognormal | n = 30 | rho = 0
## lognormal | n = 30 | rho = 0.2
## lognormal | n = 30 | rho = 0.5
## lognormal | n = 30 | rho = 0.8

result_jack_lognormal

##   distribution  n rho     var_mc   var_jack        bias          mse
## 1    lognormal 15 0.0 0.07204400 0.09552026 0.023476260 0.0137801175
## 2    lognormal 15 0.2 0.08628341 0.11263425 0.026350841 0.0156790050
## 3    lognormal 15 0.5 0.06252052 0.08966965 0.027149137 0.0101863089
## 4    lognormal 15 0.8 0.01741057 0.02887312 0.011462549 0.0019427653
## 5    lognormal 30 0.0 0.03469325 0.04315783 0.008464579 0.0038978589
## 6    lognormal 30 0.2 0.04639464 0.05746371 0.011069075 0.0049699788
## 7    lognormal 30 0.5 0.03809275 0.05139909 0.013306340 0.0037130134
## 8    lognormal 30 0.8 0.01121545 0.01598436 0.004768913 0.0004036111
```

Jackknife (Lognormal Distribution) Results

This table reports the performance of the jackknife method in estimating the variance of the Pearson correlation coefficient under a bivariate lognormal distribution, based on m = 10,000 Monte Carlo simulations

As with all previous cases, increasing the sample size from 15 to 30 leads to a decrease in the true variance (var_mc), confirming greater stability in large samples

However, the jackknife estimates (var_jack) are consistently higher than the Monte Carlo variance, showing a systematic positive bias

he bias is largest for smaller samples (n = 15), especially when the true correlation rho is moderate to high (around 0.2 to 0.5

MSE values are significantly higher than in the normal case, indicating that jackknife is less reliable dist like the lognormal distribution

Under the bivariate lognormal distribution, the jackknife method tends to overestimate the variance of the sample correlation coefficient, especially for smaller samples and moderate-to-high correlations. The bias and MSE are considerably higher than in the normal case, indicating that jackknife is sensitive to skewness and non-normality. While performance improves slightly at n = 30

Combined Jackknife Result

```
combined_jack_result <- rbind(result_jack_normal,result_jack_lognormal)

combined_jack_result
```

```
##    distribution  n rho      var_mc    var_jack         bias          mse
## 1        normal 15 0.0 0.071239191 0.082910389 0.0116711984 2.706905e-03
## 2        normal 15 0.2 0.067606369 0.076919993 0.0093136240 2.358106e-03
## 3        normal 15 0.5 0.044455711 0.052844400 0.0083886894 1.819451e-03
## 4        normal 15 0.8 0.011655757 0.015601314 0.0039455568 3.954857e-04
## 5        normal 30 0.0 0.035433400 0.036710963 0.0012775626 2.047055e-04
## 6        normal 30 0.2 0.032376530 0.034184581 0.0018080511 2.125851e-04
## 7        normal 30 0.5 0.020356800 0.022288020 0.0019312202 1.494116e-04
## 8        normal 30 0.8 0.005071503 0.005747295 0.0006757922 2.116922e-05
## 9     lognormal 15 0.0 0.072043997 0.095520257 0.0234762602 1.378012e-02
## 10    lognormal 15 0.2 0.086283413 0.112634253 0.0263508407 1.567900e-02
## 11    lognormal 15 0.5 0.062520517 0.089669655 0.0271491371 1.018631e-02
## 12    lognormal 15 0.8 0.017410573 0.028873122 0.0114625488 1.942765e-03
## 13    lognormal 30 0.0 0.034693252 0.043157831 0.0084645790 3.897859e-03
## 14    lognormal 30 0.2 0.046394638 0.057463713 0.0110690753 4.969979e-03
## 15    lognormal 30 0.5 0.038092753 0.051399094 0.0133063403 3.713013e-03
## 16    lognormal 30 0.8 0.011215446 0.015984359 0.0047689130 4.036111e-04
```

Bias & MSE Preparation

Code Explanation (Chunk: Bias & MSE Preparation)

This block of code prepares two tidy datasets — bias_data and mse_data — which will later be used for plotting or comparison between Bootstrap and Jackknife methods.

Step 1: Bias Data

From the bootstrap results (combined_boot_results), it selects:

n (sample size)

rho (true correlation)

distribution (normal/lognormal)

bias (bias of the variance estimator)

Adds a new column method = “Bootstrap”.

Repeats the same for jackknife results (combined_jack_result) but labels them "Jackknife".

Combines both into one dataset bias_data using bind_rows().

Result → A long-format dataset with bias values for each method, distribution, sample size, and ρ.

Step 2: MSE Data

From bootstrap results (combined_boot_results), it selects the same variables except mse instead of bias.

Adds method = "Bootstrap".

Repeats for jackknife results, with "Jackknife".

Combines them using rbind() into mse_data.

Result → A long-format dataset with MSE values for each method, distribution, sample size, and ρ.

This chunk reorganizes the simulation results so you can easily compare Bootstrap vs Jackknife on bias and MSE using ggplot

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
# Prepare bias data
bias_boot <- combined_boot_results %>%
  select(n, rho, distribution, bias) %>%
  mutate(method = "Bootstrap")
```

```r
bias_jack <- combined_jack_result %>%
  select(n, rho, distribution, bias) %>%
  mutate(method = "Jackknife")

bias_data <- bind_rows(bias_boot, bias_jack)

# Prepare mse data
mse_boot <- combined_boot_results %>%
  select(n, rho, distribution, mse) %>%
  mutate(method = "Bootstrap")

mse_jack <- combined_jack_result %>%
  select(n, rho, distribution, mse) %>%
  mutate(method = "Jackknife")

mse_data <- rbind(mse_boot, mse_jack)
```

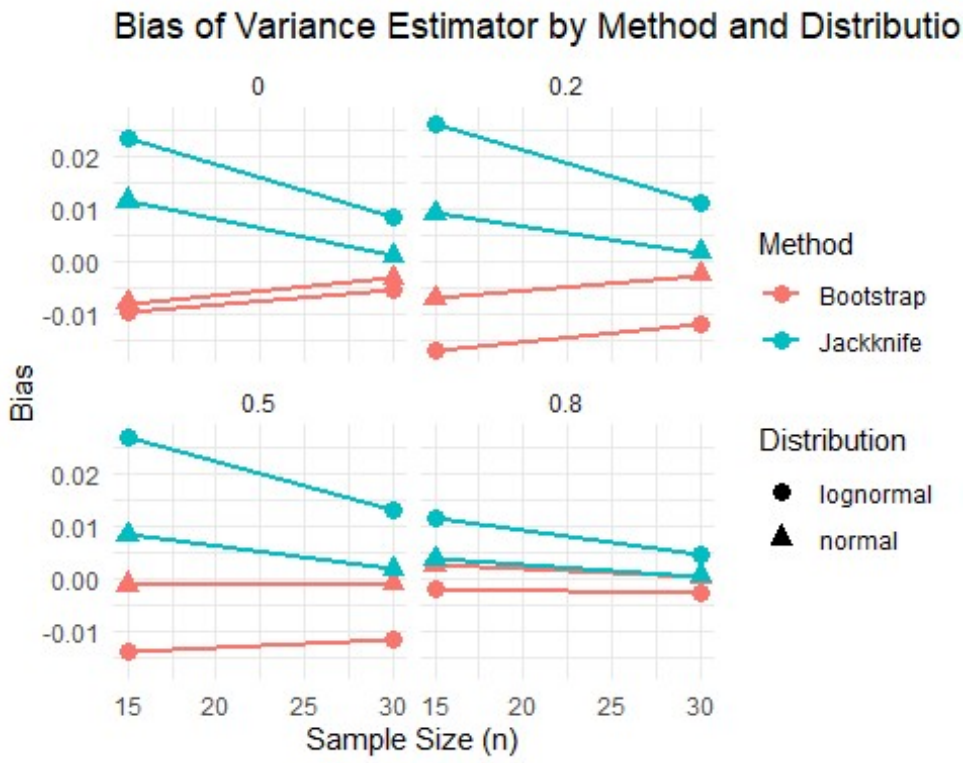
Bias Plot with ggplot2

This block of code creates a visualization of the bias values for the variance estimators, comparing Bootstrap and Jackknife across different sample sizes, correlations, and distributions.

This plot visually compares bias performance of Bootstrap vs Jackknife across different sample sizes and correlation strengths, highlighting how distribution (normal vs lognormal) influences results

```r
ggplot(bias_data, aes(x = n, y = bias, color = method, shape = distribution,
group = interaction(method, distribution))) +
  geom_line(size = 1) +
  geom_point(size = 3) +
  facet_wrap(~ rho) +
  labs(
    title = "Bias of Variance Estimator by Method and Distribution",
    x = "Sample Size (n)",
    y = "Bias",
    color = "Method",
    shape = "Distribution"
  ) +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Effect of Sample Size (n)

For Bootstrap

Bias is consistently low and stable whether n = 15 or 30,even though it has a Slight negative bias, but not extreme

For Jackknife

At n = 15, bias can be quite large positive for rho = 0, 0.2 and negative for rho = 0.5,but then reduces At n = 30,this shows that Jackknife is highly sensitive to small sample sizes, particularly for normal data

Bootstrap performs consistently well across all sample sizes. Jackknife improves with larger n but is almost unreliable for small n

Effect of Correlation

At low rho (0 and 0.2)

Jackknife shows the worst performance, especially with normal data large negative bias but Bootstrap bias remains small and stable

As rho increases to 0.5 and 0.8

ackknife bias improves,they moves close to zero,both methods become more similar in bias levels

Jackknife bias is most unstable when ρ is small. Higher correlations reduce this sensitivity. Bootstrap handles all rho values well

Effect of Distribution

Normal Distribution

Jackknife bias is much more stable under normal data Bias does not fluctuate significantly, even at small sample size (n = 15) or low correlation (rho = 0 or 0.2).

Bootstrap is also very stable, consistently showing low bias across all sample sizes and ρ levels

Lognormal Distribution

Jackknife shows larger and more variable bias, particularly when n = 15 (small sample),and rho is low (rho = 0 or 0.2)

Bootstrap remains consistently reliable, showing minimal bias for both small and large sample sizes, and across all rho values

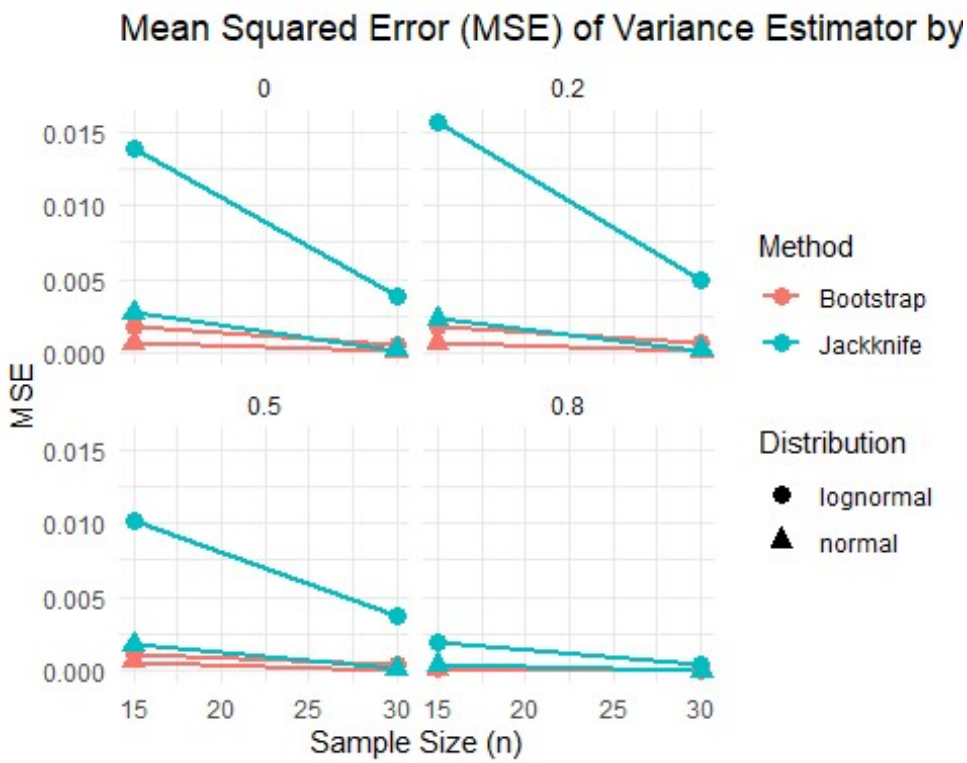
MSE Plot with ggplot2

This block generates a visualization of the Mean Squared Error (MSE) of variance estimators, comparing Bootstrap and Jackknife across different sample sizes, correlation values, and distributions.

This plot highlights how Bootstrap vs Jackknife differ in MSE performance, across sample sizes and correlation values, under both Normal and Lognormal distributions. It provides a direct visual comparison of estimator accuracy and stability

```
ggplot(mse_data, aes(x = n, y = mse, color = method, shape = distribution,
group = interaction(method, distribution))) +
  geom_line(size = 1) +
  geom_point(size = 3) +
  facet_wrap(~ rho) +
  labs(
    title = "Mean Squared Error (MSE) of Variance Estimator by Method and
Distribution",
    x = "Sample Size (n)",
    y = "MSE",
    color = "Method",
    shape = "Distribution"
```

```
  ) +
  theme_minimal()
```



Effect of Sample Size (n) on MSE

When n = 15 (small sample) The Jackknife method has significantly higher MSE, especially for lognormal data The Bootstrap method shows consistently low MSE, even at small sample sizes

When n increases to 30 MSE values decrease for both methods, but Jackknife still remains higher than Bootstrap in most cases.

The reduction is most dramatic for Jackknife under lognormal distribution, especially when rho = 0 or 0.2

Effect of Correlation (rho) on MSE

At low correlation (rho = 0 or 0.2) Jackknife struggles more, with very large MSE under lognormal, particularly at n = 15. Bootstrap remains stable and less sensitive to the correlation level.

As rho increases to 0.5 and 0.8 The MSE for Jackknife improves, especially when n = 30. Bootstrap continues to maintain a slight edge in accuracy, but the gap between methods gets smaller

Effect of Distribution Normal Distribution Both Bootstrap and Jackknife perform relatively well. Jackknife shows higher MSE at n = 15, but the difference shrinks at n = 30

Lognormal Distribution Jackknifes MSE is much larger, especially for small samples and low rho Bootstrap outperforms Jackknife clearly, with consistently lower MSE across all the rhos.

Overall, the Bootstrap method demonstrates better performance in estimating the variance of the sample correlation coefficient compared to the Jackknife, particularly in terms of bias and mean squared error (MSE). This trend holds across varying sample sizes (n = 15, 30) and correlation levels (rho = 0, 0.2, 0.5, 0.8) for both normal and lognormal distributions

The Bootstrap approach shows lower bias and MSE, showing better results even under skewed conditions like the lognormal distribution. Its performance improves steadily as the sample size increases. This indicates strong reliability

On the other hand, the Jackknife estimator is more sensitive to small sample sizes and skewed distributions. It tends to over or under estimate the variance, particularly for low rho and small n, leading to higher bias and MSE. While its performance improves slightly with larger sample sizes, it still does not match the accuracy of the Bootstrap.

Conclusion

Overall, the comparison between Bootstrap and Jackknife variance estimators for the sample correlation coefficient shows a clear advantage for Bootstrap in terms of both bias and mean squared error (MSE).

Bootstrap Strengths: Bootstrap provides consistently low bias and low MSE across all conditions small vs large sample sizes, low vs high correlations, and both Normal and Lognormal distributions. Its robustness makes it a reliable choice even under challenging conditions (e.g., skewed lognormal data or small sample sizes). The stability of Bootstrap results suggests that it adapts well regardless of correlation strength or underlying distribution.

Jackknife Weaknesses: The Jackknife estimator, in contrast, shows significant sensitivity to small sample sizes (n = 15) and skewed distributions (Lognormal). It tends to overestimate or underestimate variance depending on the correlation level, resulting in higher bias and MSE compared to Bootstrap. While its performance improves with larger samples (n = 30) and higher correlations (ρ = 0.5, 0.8), it still lags behind Bootstrap in overall accuracy and reliability.

Impact of Correlation (ρ): Jackknife is most unstable at low correlation values (ρ = 0, 0.2), particularly under Lognormal data. As ρ increases, its bias and MSE improve, but Bootstrap remains more stable throughout. This shows Bootstrap's resilience to correlation strength, while Jackknife is strongly affected.

Impact of Distribution: Under the Normal distribution, both methods perform relatively better, though Bootstrap maintains the edge. Under the Lognormal distribution, Jackknife

struggles significantly with higher variability in bias and inflated MSE, whereas Bootstrap remains consistently accurate.

Key Takeaway

Bootstrap is the preferred method for variance estimation of the correlation coefficient. It is less sensitive to small samples, performs better under skewed conditions, and maintains lower bias and MSE across all scenarios.

Jackknife can still be useful when sample sizes are large and data are normally distributed, but in small-sample or non-normal settings, it becomes unreliable.

Thus, for practical applications — especially in fields like finance or biomedical research where skewed distributions and small samples are common — Bootstrap offers a more robust and trustworthy solution.